선택적 부호화를 위한 효율적인 구문 표현

이종배[†]

요 약

동영상(Moving image) 처리시에는 연속영상(image sequence)을 프레임 단위로 압축 · 저장해두고 이를 필요에 따라 원하는 화질로 복원하여 사용하게 된다. 그런데 한 프레임 내에서도 측정한 부분들은 다른 부분들에 비해서 중요도가 큰 경우가 종종 있다. 따라서 영상을 부호화하고 전송할 때 이들 중요한 부분들의 화질을 다른 부분들과 차별화하여 부호화하는 방식이 제안되어 있으며, 향후 영상회의, MPEG-4등에서 많이 응용되리라 판단된다. 그런데 이들을 부호화하기 위해서는 중요한 부분과 그렇지 않은 부분의 경계, 텍스춰, 움직임 정보등을 표현하는 방안이 제시되어야 한다. 기존의 H.261, MPEG 등에서 제시된 방안은 각 중요한 부분과 그렇지 않은 부분들을 구별하지 않아서 본 논문에서 관심을 갖는 영상에 바로 적용하기는 곤란하다. 따라서 본 논문에서는 영상 내에 중요한 부분들이 존재하고 이들을 다른 부분들과 차별화하여 처리하고자 할 때 요구되는 특정영역의 경계, 텍스춰, 움직임 정보 등을 표현하는 방안을 제시하고자 한다.

An Efficient Syntax Rule for Selective Coding

Jong Bae Lee[†]

ABSTRACT

Image sequence is compressed and stored as the unit of frame in computer, and reconstructed with desired quality according to several applications. In some cases specific parts are more important than other parts within a frame and these important parts must be reconstructed with high quality compared with the other parts and several schemes are suggested for such application and these schemes need to separate the important parts from a given frame, and also the syntax of shape, texture, and motion information must be defined for important parts and the other parts. But syntax rule in H.261, MPEG1 or MPEG2 is not suitable for our application because the syntax rules in the existing scheme can not express shape information for separating the important regions from each frame. So we propose a new syntax rule which represents the shape, texture and motion information in the circumstances where specific important parts exist in a frame.

1. 서 론

21세기 정보화 시대를 맞이하여 각종 영상을 효율적으로 처리하기 위해서 많은 연구가 행해졌다. 영상처리는 정지영상을 다루는 경우도 있지만 주로 연속영상(image sequence)을 프레임 단위로 압축 · 저장해두고 이를 필요에 따라 원하는 화질로 복원하여 사용하게 된다. 그런데 한 프레임 내에서도 여러 형태의 물체가 혼재되어 있으며, 응용에 따라서는 이들 중에서 특정한 물체만을 골라 전송하거나, 또는 각 물체들의 화질을 차별화하여 전송할 필요가 종종 있게 된다. 일례로 전술영상에서 특별히 중요한 물체들(비행기, 군함 등)은 배경영역(수풀, 들판 등)에 비해서 그 가치가 크다고 판단되며, 화상회의 시스템에서 화상회의를 주재하는 사람은 다른 부분에 비해서 보다 중요하다고 판단된다. 또 MPEG-4[1] 시스템에서도 프레임 단위로 영상 데이터를 처리하지 않고 VO(Video Object) 단위로 부호화하여 전송하는 방식을 제안하고 있다. 이러한 방식들은 주어진 프레임을 관심영역과 배경으로 나누고, 각 영역별로 응용에

† 명지전문대학 전자계산과

맞도록 차별화하여 부호화하는 선택적 부호화 방식 [2-3]의 한 종류이다.

그런데 현재까지 제안된 H.261[4], MPEG-1[5], MPEG-2[6]등에서 제시한 각종 구문 규칙은 프레임 내에서 특정 오브젝트와 배경을 구별하지 않았으므로 선택적 부호화 방식의 구문 규칙에 바로 사용할 수가 없다. 또 특별히 선택적 부호화를 위한 구문규칙이 아직까지 제시된 예도 없다. 따라서 본 논문에서는 이점을 고려하여 선택적 부호화 방식에 적용 가능한 구문 규칙을 제시하고자 한다.

먼저 이러한 선택적 부호화를 위해서는 각 영역을 구분하기 위한 정보가 필요하다. 즉 기존의 H.261, MPEG-1, MPEG-2 등에는 없는 각 영역의 경계정보를 표현하여야 한다. 경계를 표현하기 위해서는 화소 단위로 경계를 표현하거나, 블록 단위로 경계를 표현하기도 하는데, 경계를 화소 단위로 나타내는 것은 경계를 정확하게 나타낼 수 있는 반면에 경계를 표현하기 위한 부가정보가 너무 많이 요구된다. 따라서 참고문헌[2-3]에서는 관심영역의 경계를 블록 단위로 구분하여 부호화하고 전송하는 선택적 부호화 방식을 제안하였으며 본 논문에서도 이에 맞는 부가정보 표현기법을 제시하겠다. 또 선택적 부호화 방식[2-3]에서는 블록 기반으로 영역의 경계를 구분하였기 때문에, 선택된 영역의 텍스춰 정보를 부호화할 때 기존의 H.261, MPEG-1, MPEG-2 등에서 채택하고 있는 복합 DCT/DPCM 방식을 사용하였다. 이와 같이 복합 DCT/DPCM 방식을 채택함으로써 보다 H/W 구현을 용이하게 하였으므로 본 논문에서도 DCT 계수를 부호화하는 과정 및 국부속도벡터의 Huffman 부호화 과정 등은 기존의 H.261에서 제공한 비트열 구문(bit stream syntax)을 따르겠다. 그러나 H.261에서 제시되지 않은 전역이동 파라미터, 관심영역과 배경영역 분할정보, 그룹 number 등에 대해서는 새로 비트열 구문이 정의되어야 이들을 부호화할 수 있다. 특히 방식[2-3]에서는 관심영역과 배경을 제대로 구분하기 위해서 블럭 크기를 4×4 단위까지 세분화하였으므로 이들 블럭의 오차신호의 부호화 규칙 및 경계정보 표현방안이 제시되어야 한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 각 블럭 오차신호의 부호화에 대해서 살펴보고 제 3장에서는 양자화 정보 및 보조 정보에 대한 부호화 규칙을 제시한다. 제 4장에서는 제안한 방식을 이용한 간단한 실험결과를 제시하고 결론을 맺도록 하겠다.

2. 가변블럭의 부호화

선택적 부호화란 주어진 프레임을 관심영역과 배경으로 나누고, 각 영역별로 응용에 맞도록 차별화하여 부호화하는 방식을 의미하는데, 본 논문에서는 선택적 부호화를 위해 복합 DCT/DPCM 방식을 채택하고 있다. 따라서 기존의 H.261, MPEG 등의 기존 복합 DCT/DPCM 방식처럼 각 블록의 오차신호를 부호화하기 위하여 8×8 블럭 단위로 DCT를 행하게 된다. 블럭의 크기가 8×8 이상인 경우는 8×8 블럭 단위로 나누고, 각각에 대하여 8×8 DCT 및 양자화를 행한다. 그런데 블럭의 크기가 8×8 보다 작은 경우에 8×4 DCT 장치나 4×4 DCT 장치를 이용하여 DCT를 행하면 별도의 하드웨어가 필요하다. 따라서 본 논문에서는 하드웨어 비용을 줄이기 위하여 기존의 8×8 DCT 장치를 사용하여 8×8 DCT를 행하게 되는데, 이를 위해서 8×8 보다 작은 블록들을 8×8 블럭으로 확장하는 과정이 필요하다. 블럭 확장 방식에 관하여 Chang[7] 등은 임의의 형태에 대하여 mirror 형태를 이용하여 영역을 확장하여 부호화하는 방식을 제안하였으며, Cho[8] 등은 시간 영역 및 주파수 영역에서 1차원 내삽방식(1-D interpolation method)에 의하여 블럭을 확장하는 기법을 제안하였다. 그림 1은 여러 블럭 확장방식들을 설명하고 있다. 즉 그림 1(a)의 영상 segment가 블럭 확장을 통해서, 그림 1(b), (c), (d)와 같이 된다. 먼저 zero-stuffing 기법은 그림 1(b)처럼 주어진 블럭 이외의 다른 영역에 zero를 채워넣는 가장 단순한 방식으로서 확장블럭 내에서 주어진 블럭과 zero를 채워넣는 부분 사이에 경계가 생긴다. 이 때문에 원하지 않은 고주파 성분이 많이 포함되어 부호화할 때 압축률이 많이 떨어진다. 이런 문제점을 해결하기 위하여 주어진 블럭 이외의 다른 부분에 zero가 아닌 다른 성분

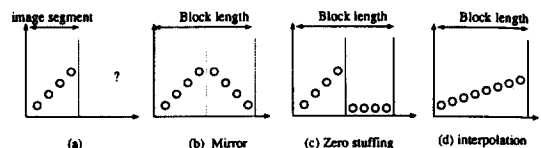



그림 1. 블럭 확장

(예를 들어 물체영역의 평균)을 채워 넣음으로서 부호화 효율을 높이는 방식이 mirror 영상 확장 방식이다. 이 방식은 그림 1(c)처럼 주어진 블럭의 경계부분을 중심으로 대칭되도록 확장 블럭에 데이터를 채워 넣는 방식이다. 이 경우 zero-stuffing 방식에 비하여 블럭 경계에서 신호의 급격한 변화를 방지하기 때문에 고주파 성분을 줄일 수 있다는 장점을 갖는다. 그러나 이들 방식은 주어진 블럭이 아닌 확장영역 부분에 원하지 않은 정보를 채워 넣기 때문에 부호화 효율이 떨어진다. 한편 공간영역에서의 선형내삽방식은 그림 1(d)처럼 주어진 블럭의 신호들을 선형 내삽하여 확장영역의 신호들을 만들어내는 방식으로, 선형 내삽된 신호를 주파수 영역으로 바꾸었을 경우, 확장된 블럭에 대응되는 부분은 거의 0의 값을 갖기 때문에 앞의 두 방식에 비해서 상대적으로 부호화 효율이 우수하다. 그런데 본 논문에서는 블럭들이 그림 2에 표시한 것과 같이 세 가지 중의 한 가지를 갖는데, 그림 2(a)는 2차원 내삽을 행하고, 그림 2(b)는 1차원 내삽을 행한다. 그림 2(c)에서는 **P**는 x-축 방향의 내삽을, **Q**는 y-축 방향의 내삽을, **R**은 인접한 화소로부터 외삽 과정을 통하여 구한다. 그림 3은 8×8 보다 작은 블럭의 부호화 과정을 블럭 다이아그램으로 표시하고 있다. 즉 8×8 블럭 확장 후에 8×8 DCT, 양자화 과정을 행한 후 zigzag scanning 및 Huffman 부호화를 행하여 복호기로 전송한다. 복호기 측에서는 반대로 역양자화 및 IDCT를 거쳐 다시 블럭 수축(block compression)을 통하여 원블럭을 재생해낸다. 2차원 영상의 블럭 확장(block expansion) 과정을 알아보기 위하여, 먼저 원 블럭(original block)의 화소를 P_{ij} ($i=0,1,\cdots,M-1$, $j=0,1,\cdots,N-1$) 로 놓고, 확장 블럭의 화소를 $C_{xy}(x=0,1,\cdots,2M-1$, $y=0,1,\cdots,2N-1)$ 로 놓자. 여기서 $M\times N$ 블럭을 내삽(interpolation)하여 $2M\times 2N$ 블럭으로 확장시킬 경우, 식 1에 의해 확장 블럭의 좌표 C_{xy}가 계산되고, 역으로 P_{ij}는 식 2에 의해 표시된다.

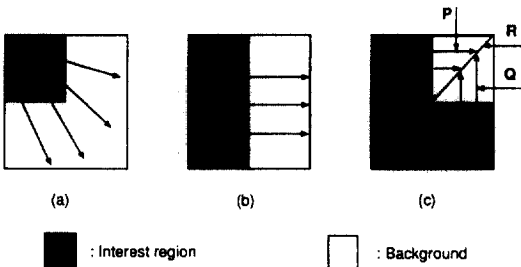
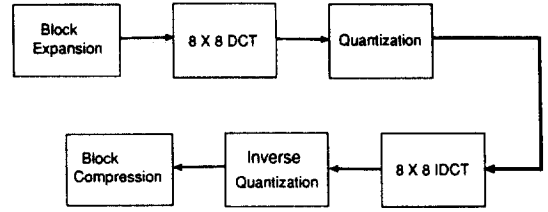



그림 2. 블럭 확장의 예

그림 3. 8×8 보다 작은 블럭의 부호화

$$\begin{aligned} C_{xy} = {} & P_{ij}\left(i+1-\frac{M-1}{2M-1}x\right)\left(j+1-\frac{N-1}{2N-1}\right)+ \\ & P_{i(j+1)}\left(i+1-\frac{M-1}{2M-1}x\right)\left(\frac{N-1}{2N-1}y-j\right)+ \\ & P_{(i+1)j}\left(\frac{M-1}{2M-1}x-i\right)\left(j+1-\frac{N-1}{2N-1}y\right)+ \\ & P_{(i+1)(j+1)}\left(\frac{M-1}{2M-1}x-i\right)\left(\frac{N-1}{2N-1}y-j\right) \end{aligned} \quad (1)$$

여기서 $i = [\frac{M-1}{2M-1}x]$, $j = [\frac{N-1}{2N-1}y]$ 로 표시되며, [*]의 의미는 *를 넘지 않은 최대 정수를 말한다.

$$\begin{aligned} P_{ij} = {} & C_{xy}\left(x+1-\frac{2M-1}{M-1}i\right)\left(y+1-\frac{2N-1}{N-1}j\right)+ \\ & C_{x(y+1)}\left(x+1-\frac{2M-1}{M-1}i\right)\left(\frac{2N-1}{N-1}j-y\right)+ \\ & C_{(x+1)y}\left(\frac{2M-1}{M-1}i-x\right)\left(y+1-\frac{2N-1}{N-1}j\right)+ \\ & C_{(x+1)(y+1)}\left(\frac{2M-1}{M-1}i-x\right)\left(\frac{2N-1}{N-1}j-y\right) \end{aligned} \quad (2)$$

여기서 $x = [\frac{2M-1}{M-1}i]$, $y = [\frac{2N-1}{N-1}j]$이다.

마찬가지로 $M\times 2N$ 블럭을 내삽(interpolation)하여 $2M\times 2N$ 블럭으로 확장시킬 경우, 식 (3)에 의해서 확장 블럭의 좌표 C_{xy}가 계산되고, 역으로 원 블럭의 좌표 P_{ij}는 식 (4)에 의해서 구해진다.

$$C_{xy} = P_{ij}\left(i+1-\frac{M-1}{2M-1}x\right)+P_{(i+1)j}\left(\frac{M-1}{2M-1}x-i\right) \quad (3)$$

여기서 $i = [\frac{M-1}{2M-1}x]$, $j = y$ 이다.

$$P_{ij} = C_{xy}(x+1-\frac{2M-1}{M-1}i) + C_{(x+1)y}(\frac{2M-1}{M-1}i-x) \quad (4)$$

여기서 $x = [\frac{2M-1}{M-1}i]$, $y = j$ 이다.

2.1 4×4 블럭 및 4×8 블록의 8×8 블럭으로의 확장

식 (1), 식 (2)를, '4×4 블럭의 8×8 블럭으로의 확장'에 적용하면, 각각 식 (5), 식 (6)과같다.

$$C_{xy} = P_{ij}(i+1-\frac{3}{7}x)(j+1-\frac{3}{7}y) + P_{i(j+1)}(i+1-\frac{3}{7}x)(\frac{3}{7}y-j) + P_{(i+1)j}(\frac{3}{7}x-i)(j+1-\frac{3}{7}y) + P_{(i+1)(j+1)}(\frac{3}{7}x-i)(\frac{3}{7}y-j) \quad (5)$$

여기서 $i = [\frac{3}{7}x]$, $j = [\frac{3}{7}y]$이다.

$$P_{ij} = C_{xy}(x+1-\frac{7}{3}i)(y+1-\frac{7}{3}j) + C_{x(y+1)}(x+1-\frac{7}{3}i)(\frac{7}{3}j-y) + C_{(x+1)y}(\frac{7}{3}i-x)(y+1-\frac{7}{3}j) + C_{(x+1)(y+1)}(\frac{7}{3}i-x)(\frac{7}{3}j-y) \quad (6)$$

여기서 $x = [\frac{7}{3}i]$, $y = [\frac{7}{3}j]$이다.

마찬가지로 4×8 블럭을 내삽(interpolation)하여 8×8 블럭으로 확장시킬 경우, 식 (7), 식 (8)에 의해서 확장 블럭의 좌표 C_{xy} 및 원 블럭의 좌표 P_{ij}가 계산된다.

$$C_{xy} = P_{ij}(i+1-\frac{3}{7}x) + P_{(i+1)j}(\frac{3}{7}x-i) \quad (7)$$

여기서 $i = [\frac{3}{7}x]$, $j = y$이다.

$$P_{ij} = C_{xy}(x+1-\frac{7}{3}i) + C_{(x+1)y}(\frac{7}{3}i-x) \quad (8)$$

여기서 $x = [\frac{7}{3}i]$, $y = j$이다.

2.2 그림 2(c)의 8×8 블럭으로의 확장

이 경우 여러 가지의 블럭 확장을 생각할 수 있으나, 본 논문에서는 다음과 같이 확장을 행한다.

Case 1 : $0 \le x \le 3$ & $0 \le y \le 3$

$$C_{xy} = P_{ij} \quad (9)$$

여기서, $i = x$, $j = y$ 이다.

Case 2 : $0 \le x < y$ & $4 \le y \le 7$

$$C_{xy} = (i+1-\frac{3}{y-1}x)P_{ij} + (\frac{3}{y-1}x-i)P_{(i+1)j} \quad (10)$$

여기서, $i = [\frac{3}{y-1}x]$, $j = y$ 이다.

Case 3 : $4 \le x \le 7$ & $0 \le y < x$

$$C_{xy} = (j+1-\frac{3}{x-1}y)P_{ij} + (\frac{3}{x-1}y-j)P_{i(j+1)} \quad (11)$$

여기서, $i = x$, $j = [\frac{3}{x-1}y]$ 이다.

Case 4 : $x = y$ $(4 \le x, y \le 7)$

$$C_{xy} = C_{(x-1)y} + C_{x(y-1)} - 0.5C_{(x-2)y} - 0.5C_{x(y-2)} \quad (12)$$

역으로 P_{ij}의 재생은 아래와 같다.

Case 1 : $0 \le i \le 3$ & $0 \le j \le 3$

$$P_{ij} = P_{xy} \quad (13)$$

여기서, $x = i$, $y = j$ 이다.

Case 2 : $0 \le i \le 3$ & $4 \le j \le 7$

$$P_{ij} = (x+1-\frac{3}{j-1}i)C_{xy} + (\frac{3}{j-1}i-x)C_{(x+1)y} \quad (14)$$

여기서, $x = [\frac{3}{j-1}i], y = j$ 이다.

Case 3 : $4 \le i \le 7$ & $0 \le j \le 3$

$$P_{ij} = (y+1-\frac{3}{i-1}j)C_{xy} + (\frac{3}{i-1}j-y)C_{x(y+1)} \quad (15)$$

여기서, $x = i, y = [\frac{3}{i-1}j]$ 이다.

3. 전송비트 할당 규칙

본 절에서는 주어진 프레임에 대하여, 블럭 오차 신호 및 움직임 파라미터 정보, 분할 정보를 어떻게 비트 스트림으로 변환시켜서 전송하는 지를 나타내고 있다. 블럭 오차신호는 기본적으로 H.261이나 H.263[9]에서 사용한 DCT 및 양자화기를 사용하므로, 비트열을 만드는 과정 및 비트열로부터 원래의 신호를 복원해내는 과정은 동일하다. 단지 양자화 과정에서 배경과 관심영역에 대한 QP 값을 다르게 설정하므로 추가로 배경영역과 관심영역에 대한 정보 및 해당 QP 값이 전송되어야 한다. 즉 복호기측에 해당 블럭이 배경영역인지, 관심영역인지, 화면간 부호화 블럭(inter mode)인 지, 화면내 부호화 블럭(intra mode)인 지 등을 알려주어야 제대로 원래신호를 복원해 낼 수 있다. 한편, H.261에서 최대 양자화 파라미터 값(QP=31)을 사용할 수 없을 정도로 버퍼 범람의 우려가 있을 경우에는, 일반적으로 이전영상의 블럭 데이타를 그대로 사용하여 주어진 블럭을 재생하므로 해당 블럭의 화질 열화가 심하다. 따라서 제안한 방식에서는 화면내 부호화 블럭의 경우에는 DC 값 만을 전송해주는 모드를 두고, 화면간 부호화 블럭의 경우에는 과거영역 데이터와 미리 전송된 분할정보, 움직임 정보들을 이용하여 주어진 블럭의 신호를 재생해 낼 수 있도록 하여 영상 내의 급격한 화질 열화를 줄일 수 있도록 하였다. 블럭오차신호를 제외한 나머지 신호를 보조정보라 정의하는데, 이들에 대한 비트 할당을 살펴보자. 보조정보는 크게 각 프레임 header 정보, 전역이동 파라미터 및 지역이동 정보, 영역분할정보 등으로 나눌 수 있다. 프레임 header 정보는 주어진 프레임의 부호화 시작을 알려주는 프레임 start, 배경영역과 관심영역에 대한 양자화 파라미터등 각 프레임 단위로 결정되는 파라미터들을 의미한다. 전역이동 파라미터는 파라미터의 갯수 및 각 파라미터 정도(resolution)에 따라 발생 비트량이 달라진다. 지역이동정보의 경우, H.263의 경우처럼 half-pel 단위까지 측정하였다. 움직임 벡터에 대하여 가변길이부호(variable length code ; VLC) 비트할당은 H.263[4]에서 사용한 테이블을 그대로 사용하였다. 지역움직임의 최대 이동거리는 응용 영상에 따라 조절될 수 있는데, H.263 테이블에 표시되지 않은 값은 13 비트 이상으로 VLC를 구성하여 사용한다. 제안된 부호화 방식의 경우는 관심영역과 배경영역의 분할을 행하므로 영역분할 정보가 어떤 형태로든 복호기측으로 전송되어야 한다. 분할정보 전송을 위해서는 먼저 분할 정보를 적절히 비트열로 표현하여야 하는데, 분할정보를 표현하는 방식으로는 크게 chain coding, run length coding, quadtree 방식등을 생각할 수 있다. Multi-color를 표현하기 위한 quadtree 방식은 여러 가지가 있으나 본 논문에서는 depth first search(DFS) 방식과 compact complementary quadtree(CCQ) 방식[10]을 이용한 실험 결과를 제시하겠다. DFS 방식은 이진영상에 대하여 가장 일반적인 quadtree 표현방식의 하나로 본 논문에서는 그림 4와 같이 이를 multi-color를 표현할 수 있도록 확장하였다. DFS 방식은 먼저 각 영역의 균질성을 판별하여 주어진 영역이 균질할 경우는 leaf로 정의하고 leaf 정보 0과 그 영역의 label 정보를 보낸다. 만약 해당영역이 균질하지 않을 경우는 node 정보 1을 전송하고 주어진 영역을 다시 네 개의 부영역으로 분할한다. 한편 CCQ 방식이 그림 5에 제시되어 있는데, 각 node에는 주어진 영역의 dominant 정보(주어진 영역에서 가장 많은 화소를 갖는 label)를 갖고 있으므로 부호화시 쉽게 tree level을 바꿀 수 있어서 발생 데이타량의 조절이 용이하다. 그림 4(a)의 영상을 CCQ 방식으로 부호화한 결과가 그림 5(b)에 제시되어 있다. 즉 그림 4(a)는 그림 6(a)에서 정의한 type code와 non-dominant 정보(주어진 영역에서 dominant 정보가 아닌 leaf label)에 의해 표

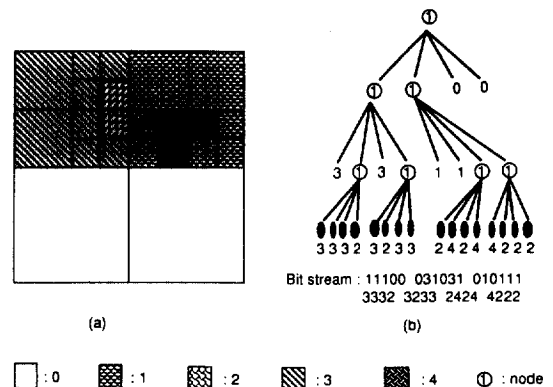



그림 4. multi-color 정보의 DFS 방식을 이용한 quadtree 표현

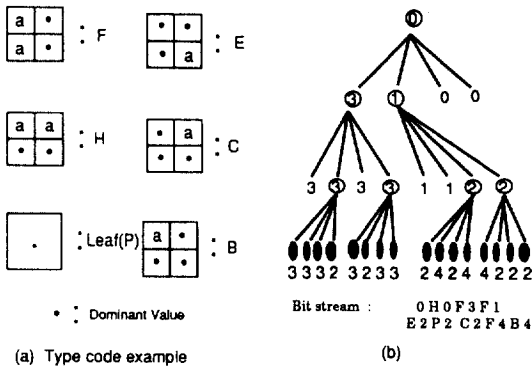



그림 5. multi-color 정보의 CCQ 방식을 이용한 quadtree 표현

그림 6. 4×4 블럭 영상

시되며, 그림 5(b)가 이들의 tree 구조 및 비트열을 나타내고 있다.

4. 실험결과 및 결론

그림 6은 4×4 블럭을 실험하기 위하여 Table tennis 첫번째 프레임을 변형한 영상으로, 8×8 블럭으로부터 한 개의 서브블럭을 선택하여 모의실험을 행하였다. 4×8 블럭이나 그림 2(c)의 경우도 비슷하게 블럭을 변형하여(8×8 블록에 대해서 4×8 블록, 또는 그림 2(c)와 같은 형태의 서브블럭을 한 개 선택함) 모의실험을 행하였다. 표 1은 Table tennis 연속영상에 대하여, 블럭의 내삽방식(interpolation method), zero stuffing 방식 및 mirror 확장 방식을

표 1. 각 블럭 확장 방식의 모의실험 (PSNR(dB)/사용비트) **실험영상** : Table tennis 연속영상

TYPE	Zero Stuffing	Mirror	Interpolation
Fig. 6(a)	22.33/125054	29.10/46149	32.68/44014
Fig. 6(b)	26.05/118501	29.29/68931	31.17/66946
Fig. 6(c)	24.03/128567	27.11/66047	27.37/64598

이용한 실험 결과를 나타내고 있다. 여기서 PSNR 및 사용된 비트는 Table tennis 연속영상 첫 번째 프레임으로부터 다섯 번째 프레임까지 수행한 결과의 평균값을 의미한다. 즉 세 방식을 비교하여 보면, zero stuffing의 경우 다른 방식들에 비해 많은 비트가 할당되었음에도 불구하고 PSNR 측면에서 다른 두 방식에 비하여 화질이 많이 떨어짐을 알 수 있다. mirror 확장방식의 경우에도, 내삽방식에 비하여 비트가 더 많이 사용되었지만, 내삽방식에 비하여 PSNR이 많이 떨어짐을 알 수 있다. 특히 4×4 블럭 확장의 경우에, 내삽 방식이 우수한 성능을 나타냈는데, 이는 블럭 내의 중요하지 않는 부분이 그림 2 (b)나 (c)의 경우에 비해 많고, 내삽 방식의 경우 2-D 내삽 확장이 가능하기 때문이다.

본 논문에서는 2 가지 분할정보 영상을 만들게 되는데, 하나는 관심영역/배경, 화면간 부호화/화면내 부호화 블럭을 나타내는 영상이고, 다른 하나는 group number를 표시하는 영상으로서, 그림 7, 그림 8은 'Table tennis 연속영상' 첫 번째와 두 번째 프레임에

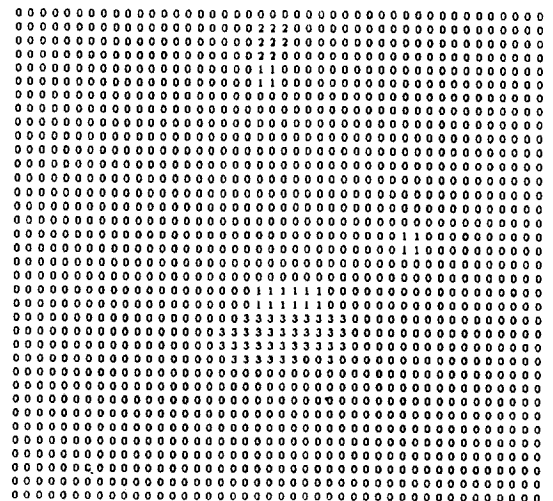



그림 7. 배경/관심영역, 화면간 부호화/화면내 부호화 블럭 표현 영상

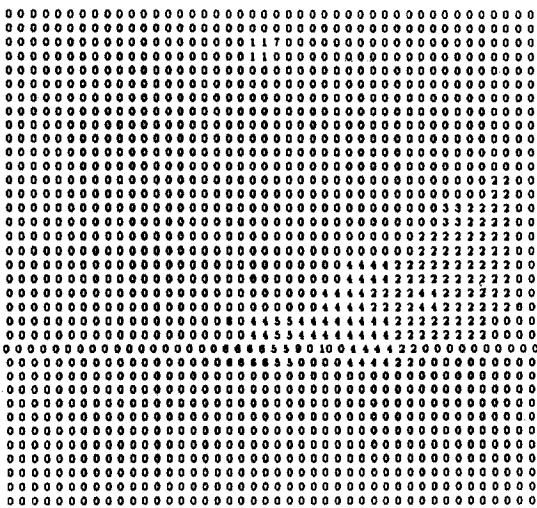



그림 8. 그룹 number 표현 영상

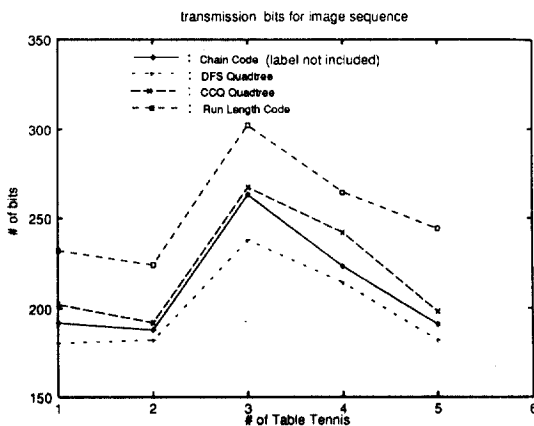



그림 9. 영역 분할 정보 표현시 비트 발생량

대해 각각 '관심영역/배경, 화면간 부호화/화면내 부호화 블럭 영상(QUAD_1)', 'group 표현 영상(QUAD_2)'을 나타내고 있다. QUAD_1을 구성하기 위해서는 관심영역/배경영역, 화면간 부호화/화면내 부호화 블럭을 결정하여야 하는데, 그림 7이 이들 정보를 표시하고 있다. 그림 7은 'Table tennis 연속영상' 첫번째와 두번째 프레임 사이에서 관심영역들이 탁구공(중요도가 가장 큰 영역)과 탁구 라켓(중요도가 두번째인 영역) 두 가지인 경우 주어진 블럭이 배경영역인 지, 어떤 관심영역인 지, 화면내 부호화 블럭이 지 화면간 부호화 블럭인 지를 표시한다. 그림 8는 그룹 number를 나타내는 영상으로서, 국부속도벡터의 대표값을 나타내고 있다.

이들 영상에 대한 영역분할 정보를 표시하기 위하여 본 방식에서는 quadtree 방식을 이용하였는데 그 이유는 다음과 같다. 먼저 방식[2-3]의 이동영역 분할 과정에서 quadtree 방식이 이용되므로, H/W 구현 측면에서 chain coding이나 run length coding 등의 방식에 비해 보다 용이하다. 그림 9는 비트 발생 측면에서 여러 방식을 비교한 결과로서 DFS quadtree 방식이 가장 성능이 좋은데, 이는 분할된 영상의 구조가 그 주된 이유이다. 즉, 본 논문에서는 그림 7이나 8과 같은 분할정보 영상을 만들게 되는데, 이들 모두 여러 분할된 영역이 존재하므로 chain coding의 경우 각각의 분할된 영역에 대해 초기 절대좌표를 전송해 주어야 하고, 이들이 비트 증가의 주요한 원인이 된다. 또 run-length coding의 경우, label 값 및 run length를 보내 주어야 하는데, 본 방식의 경우 label이 많고 주사선 방향으로 label의 변화 횟수가 많으므로 비효율적이다. 한편 quadtree 방식에 있어서 DFS 방식이 CCQ 방식에 비해 발생 비트량이 적은데, 이는 DFS 방식에서는 node level이 최소가 아닌 한 배경영역에 대한 label 정보를 전송할 필요가 없는 반면에, CCQ 방식은 그렇지 못하기 때문이다. 따라서 제안한 부호화 시스템의 경우 배경영역이 전체영상에서 대부분을 차지하는 경우가 많으므로 DFS 방식을 사용하여 영역분할 정보를 표현한다. 또 다른 quadtree 방식의 장점으로는 quadtree 정보에 각 영역의 위치 정보뿐만이 아니라, label 정보도 포함하고 있기 때문에 따로 label 정보를 전송해 줄 필요가 없다.

참 고 문 헌

[1] AOE Group, "Coding of moving pictures and associated audio : MPEG-4 Committee Draft", *ISO/IEC JTC1/SC29/WG11, N2202*, Mar. 1998.

[2] J. B. Lee and S. D. Kim, "Selective Coding Scheme Based on Global/Local Motion Information," *Signal Process., Vol. 55, Dec. 1996*, pp. 339-350.

[3] J. B. Lee and S. D. Kim, "Selective Coding Scheme for Reconstructing an Interest Region

with High Quality", *IEICE Trans. Fundamentals, Vol. E81-A, No.1 Jan. 1998,* pp. 183-191.

[4] "Draft revision H.261," *CCITT SGXV*, March 1990.

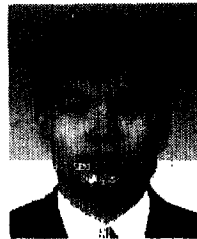
[5] Simulation Model Editorial Group, "MPEG Video Simulation Model Three(SM3)," *ISO/IEC JTC1/SC29/WG11, MPEG90/No.041,* July 1990.

[6] MPEG Video Editorial Group, "Generic coding of moving pictures and associated audio : ISO/IEC 13818-2 Draft International Standard," *ISO/IEC JTC1/SC29/WG11, MPEG94/No. 702,* Mar. 1990.

[7] S. F. Chang and D. G. Messerschmitt, "Transform Coding of Arbitrary-Shaped Image Segments", *ACM Multimedia, Vol. 7,* Jul. 1985, pp. 384-401.

[8] C. S. Cho and S. D. Kim, "Texture coding using 2-D DCT based on extension interpolation," *IEICE Trans. Fundamentals, Vol. E80-A, E80-A,* pp.789-793, 1997.

[9] "Draft Recommendation H.263", ITU-TS SG15 Working Party 15/1, March 1995.

[10] 김영모, "상보 쿼드트리에 의한 영상의 계층적 방법," *한국과학기술원 박사학위 논문,* 1989.

이 종 배

1985년 2월 서울대학교 제어계측학과 졸업(학사)
1987년 2월 KAIST 전기 및 전자공학과 졸업(석사)
1996년 2월 KAIST 전기 및 전자공학과 졸업(박사)
1985년 3월~1996년 9월 LG전자 연구원
1996년 10월~1997년 8월 특허청 영상기기과 심사관
1997년~현재 명지전문대학 전자계산과 조교수